

FIG. 1

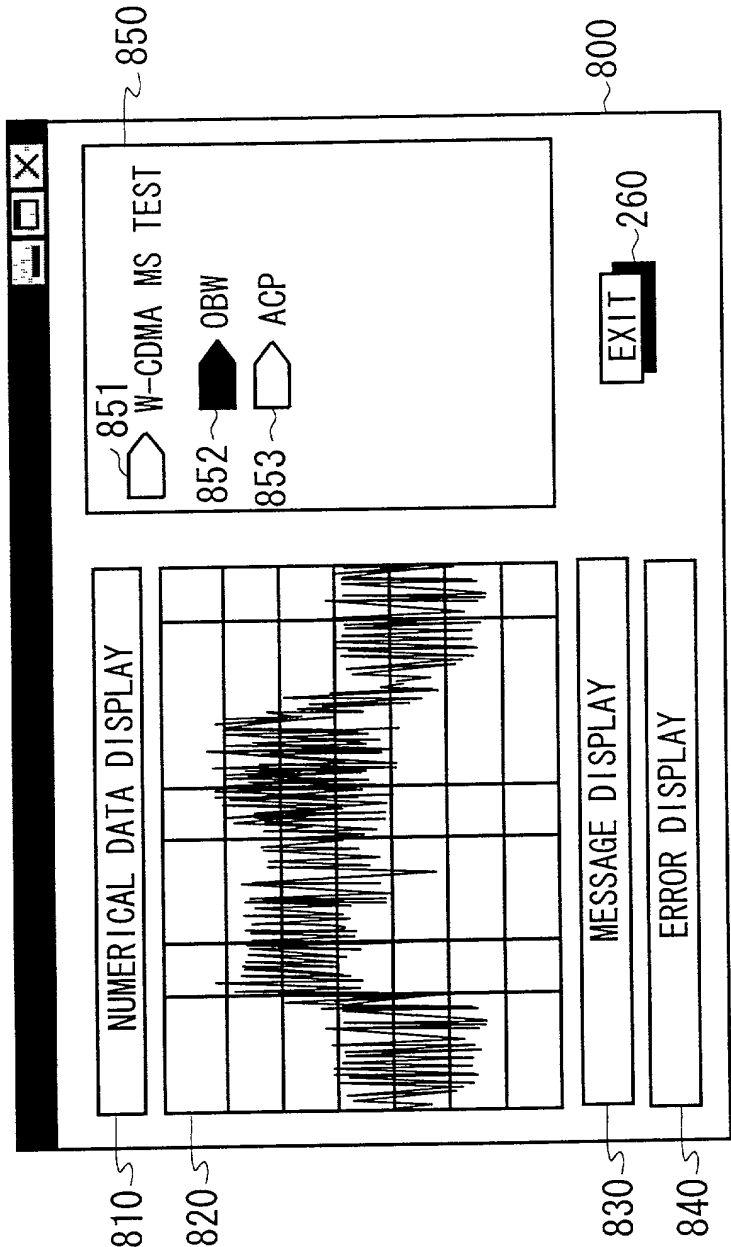


FIG . 2

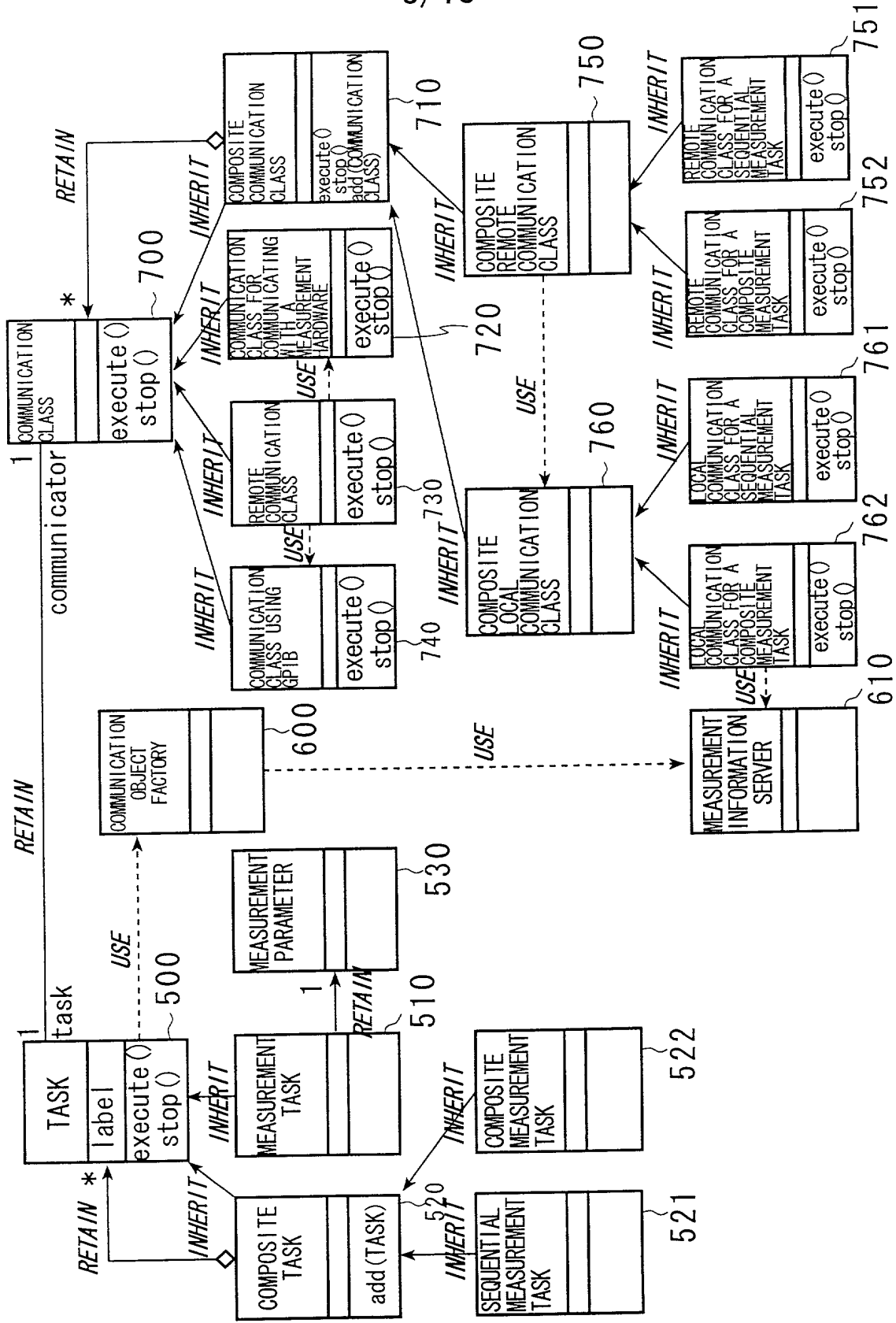


FIG. 3

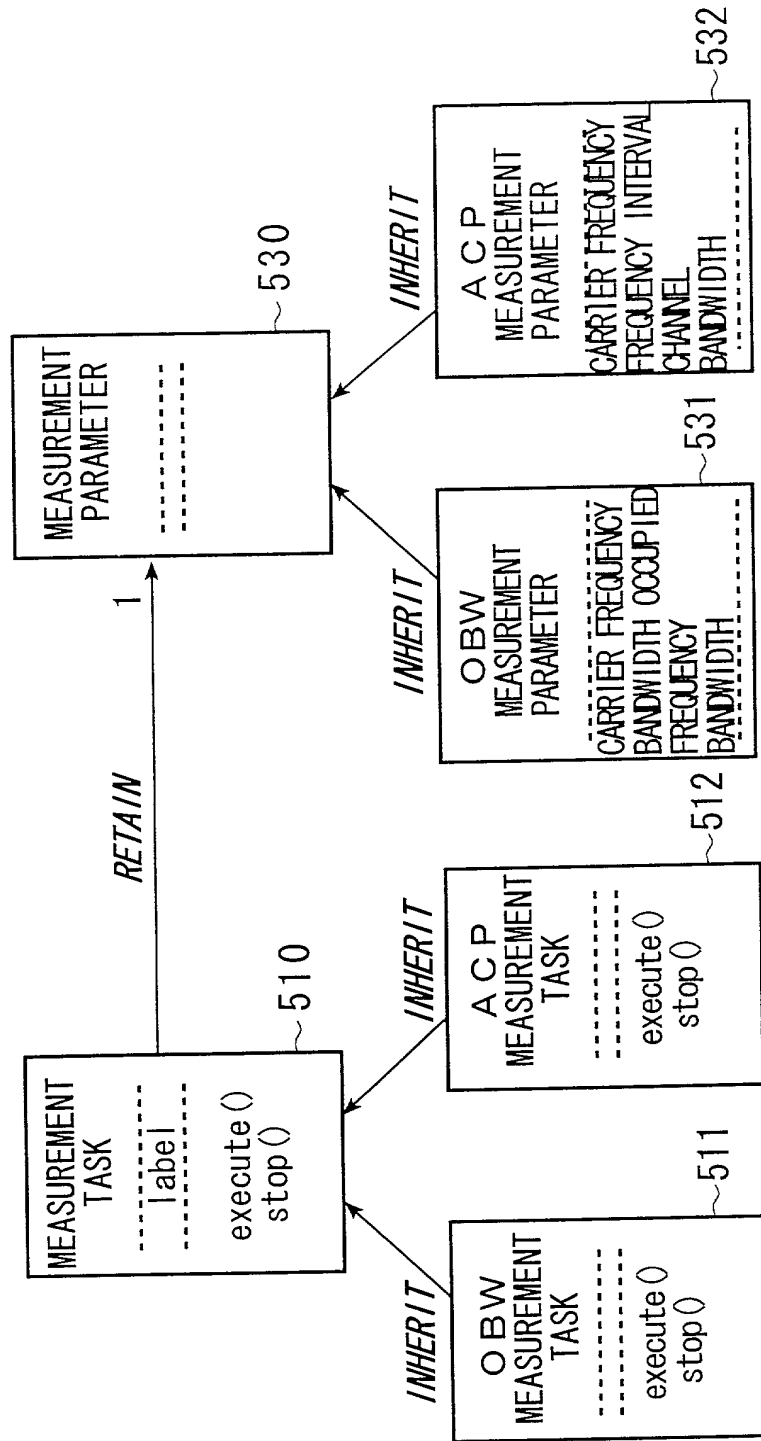


FIG. 4

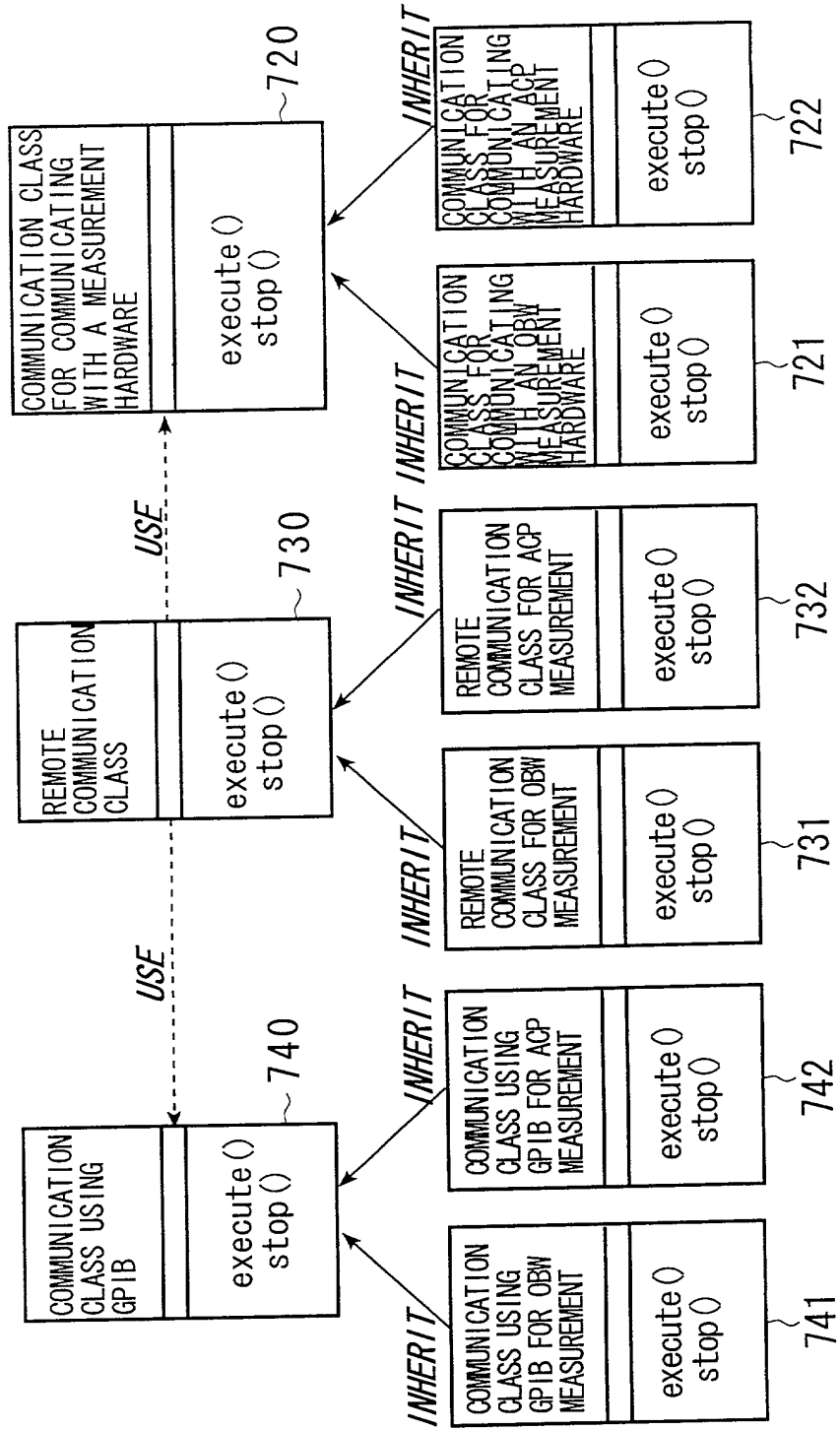


FIG. 5

```

0:  try{
1:      GENERATE COMPOSITE MEASUREMENT TASK
2:      GENERATE O B W MEASUREMENT TASK
3:      SETUP MEASUREMENT PARAMETER ON O B W MEASUREMENT TASK
4:      GENERATE A C P MEASUREMENT TASK
5:      SETUP MEASUREMENT PARAMETER ON A C P MEASUREMENT TASK
6:      ADD O B W MEASUREMENT TASK TO RETAINING RELATION OF COMPOSITE MEASUREMENT TASK
7:      ADD A C P MEASUREMENT TASK TO RETAINING RELATION OF COMPOSITE MEASUREMENT TASK
8:  }
9:  catch (STORE TASK EXCEPTION INFORMATION TO VARIABLE E) {
10:      TASK EXCEPTION HANDLING
11:  }

```

FIG. 6

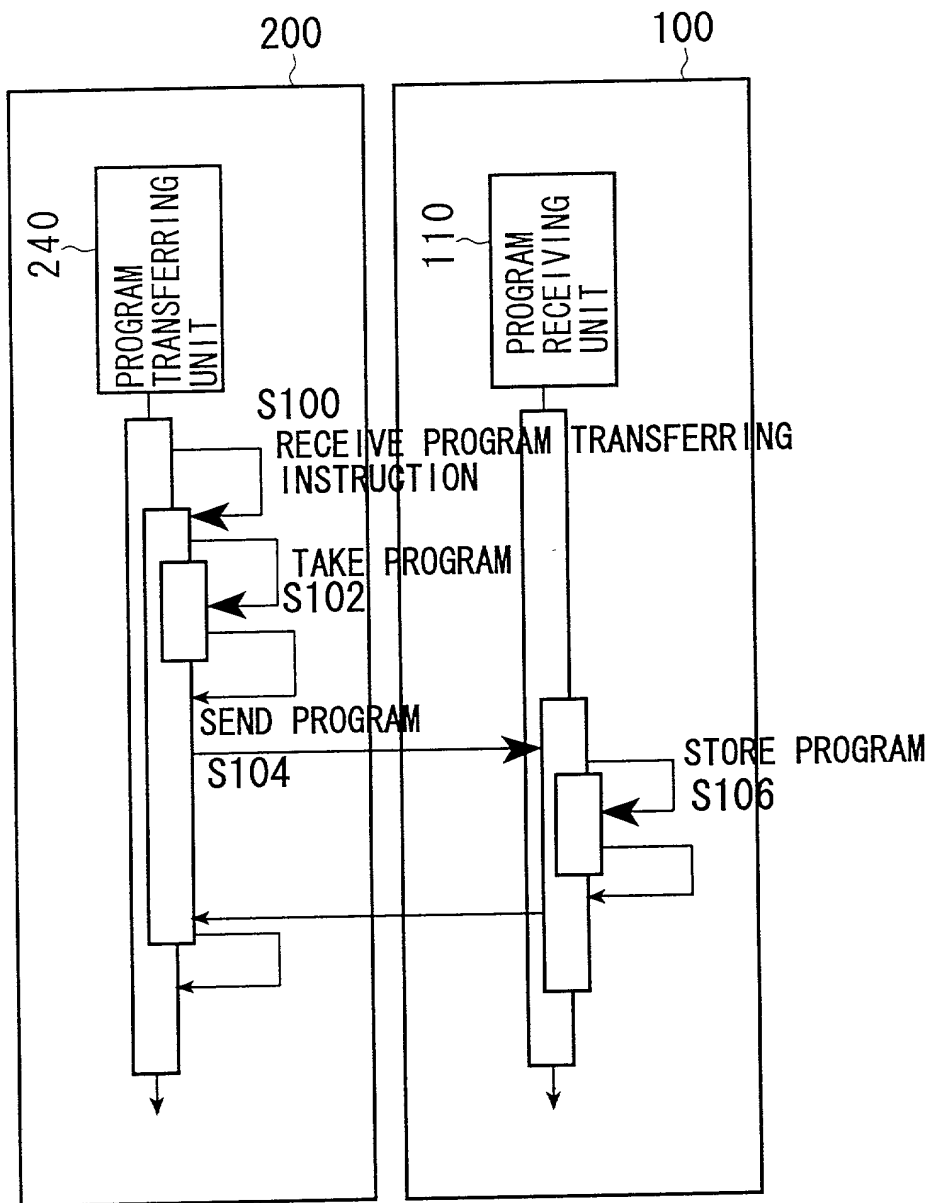


FIG. 7

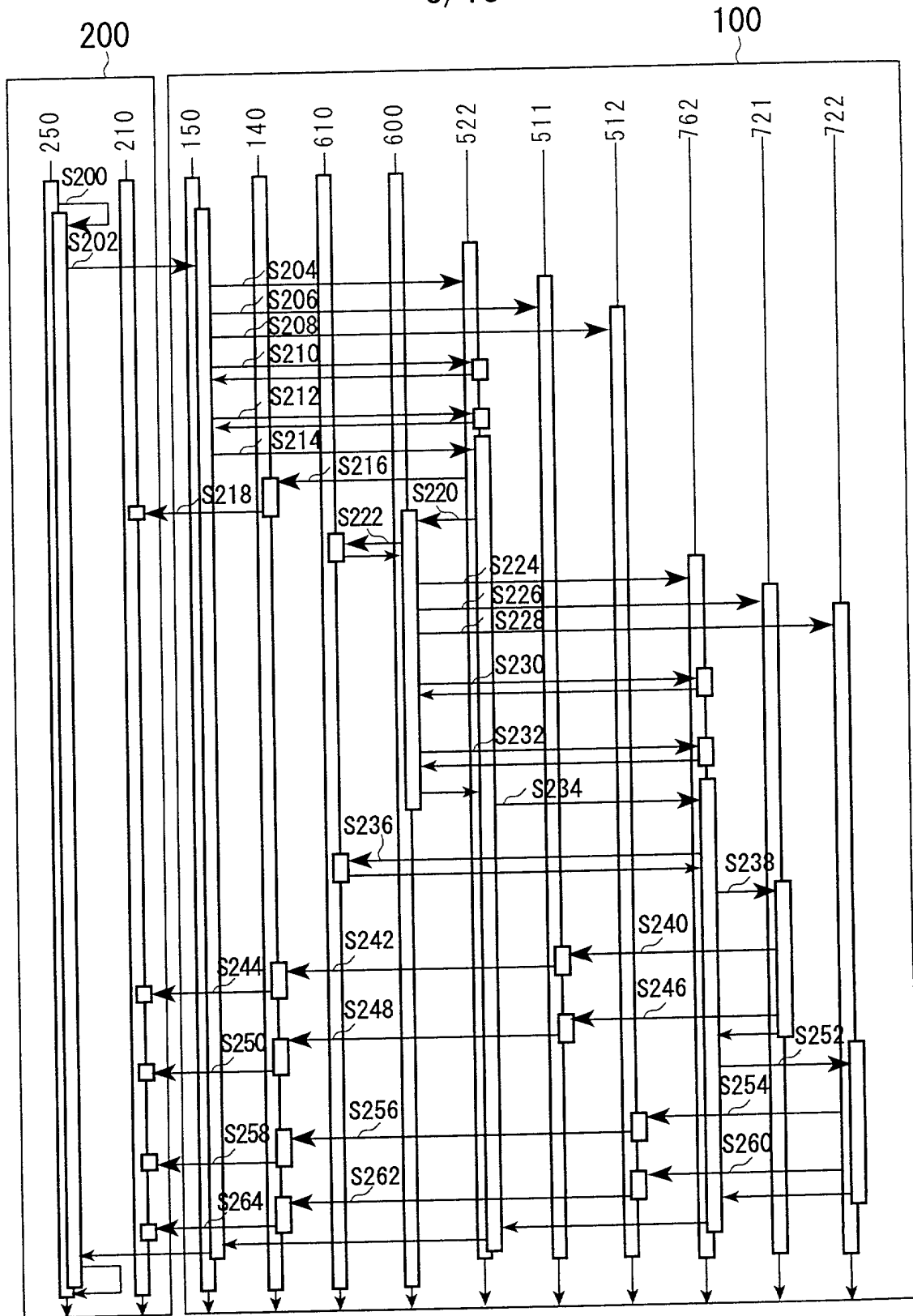


FIG. 8

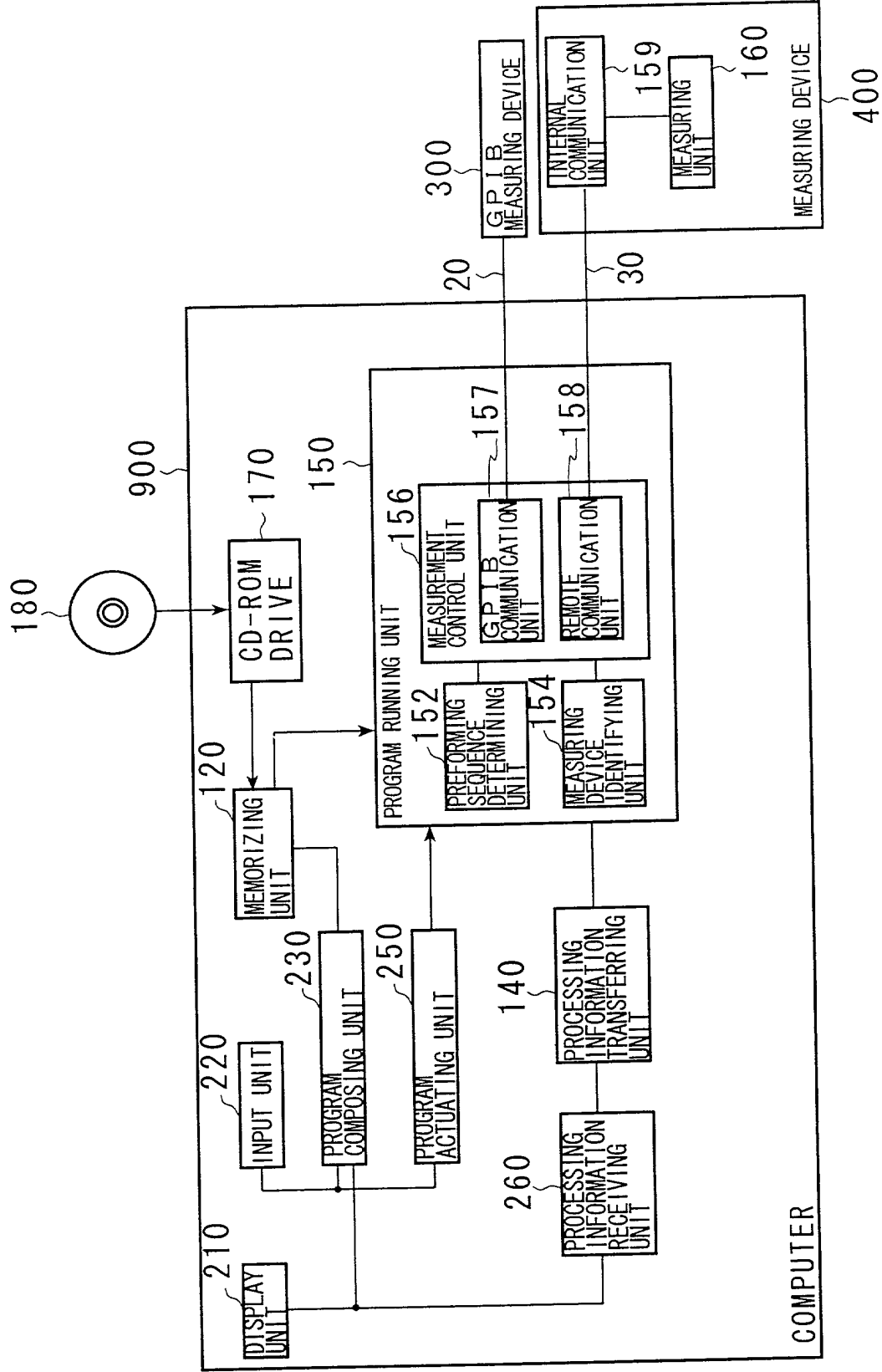


FIG. 9

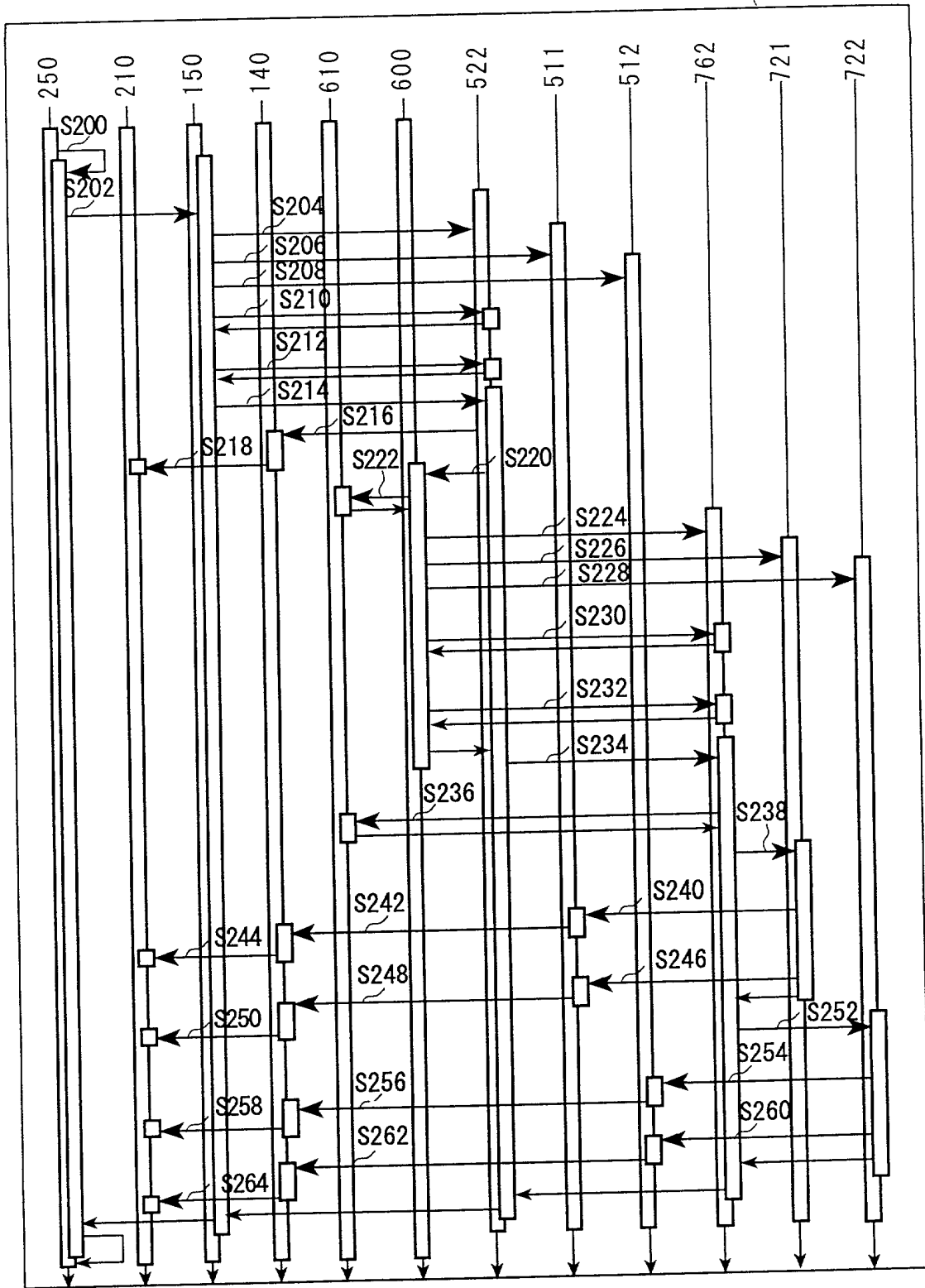


FIG. 10

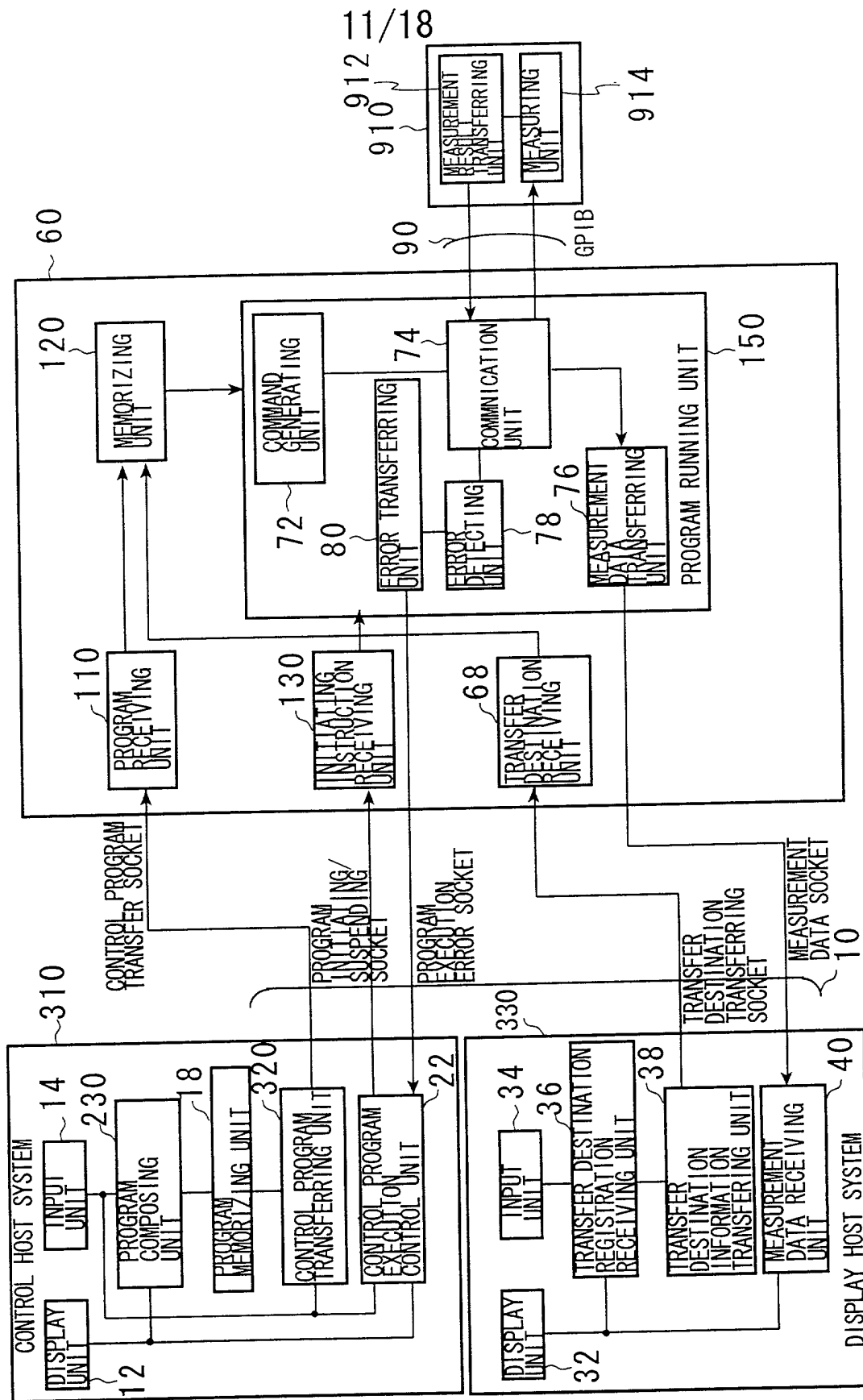


FIG. 11

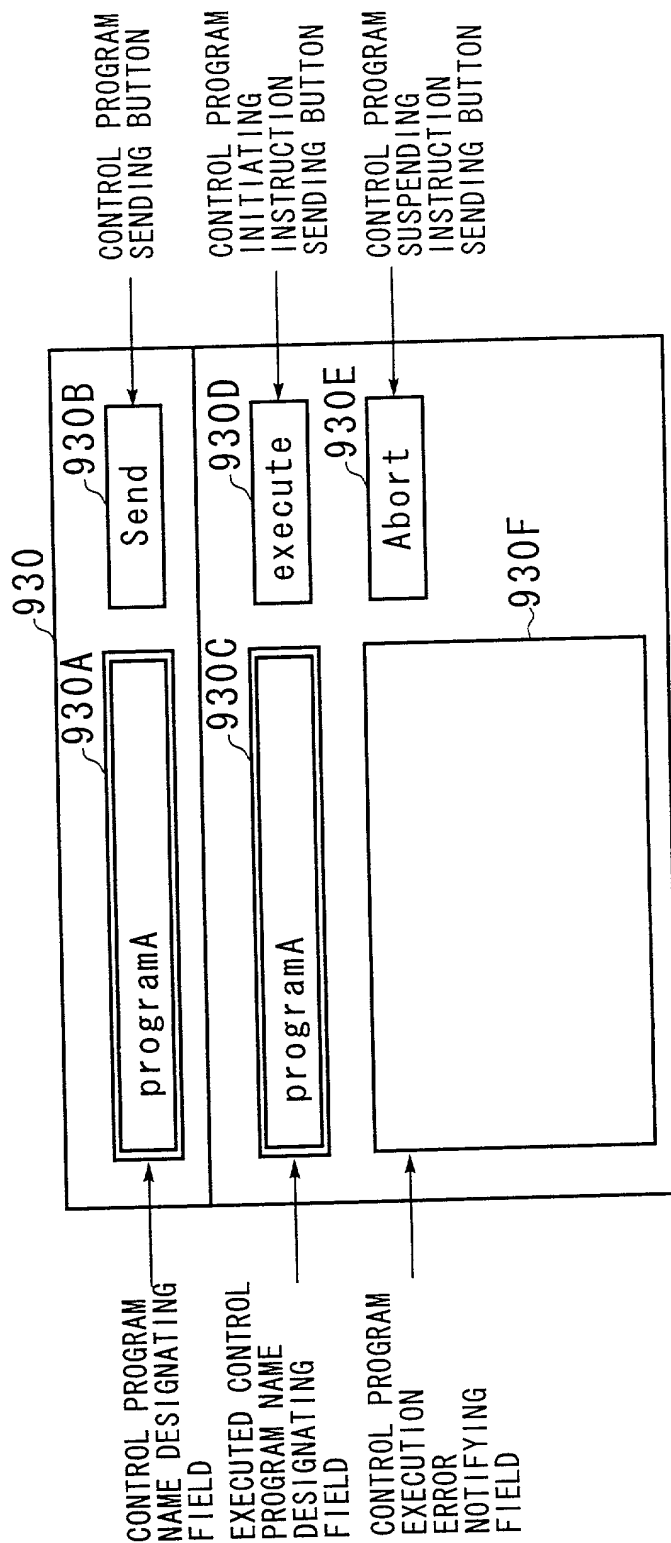


FIG . 12

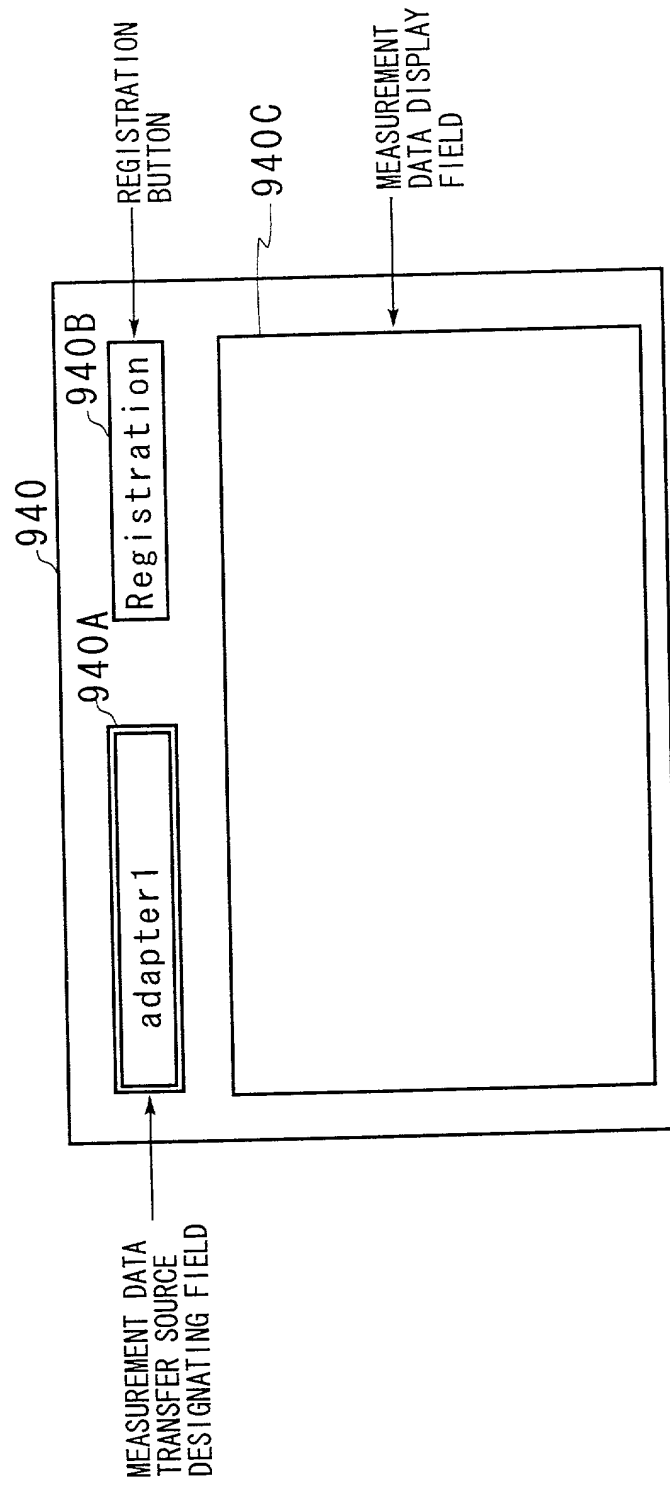


FIG . 13

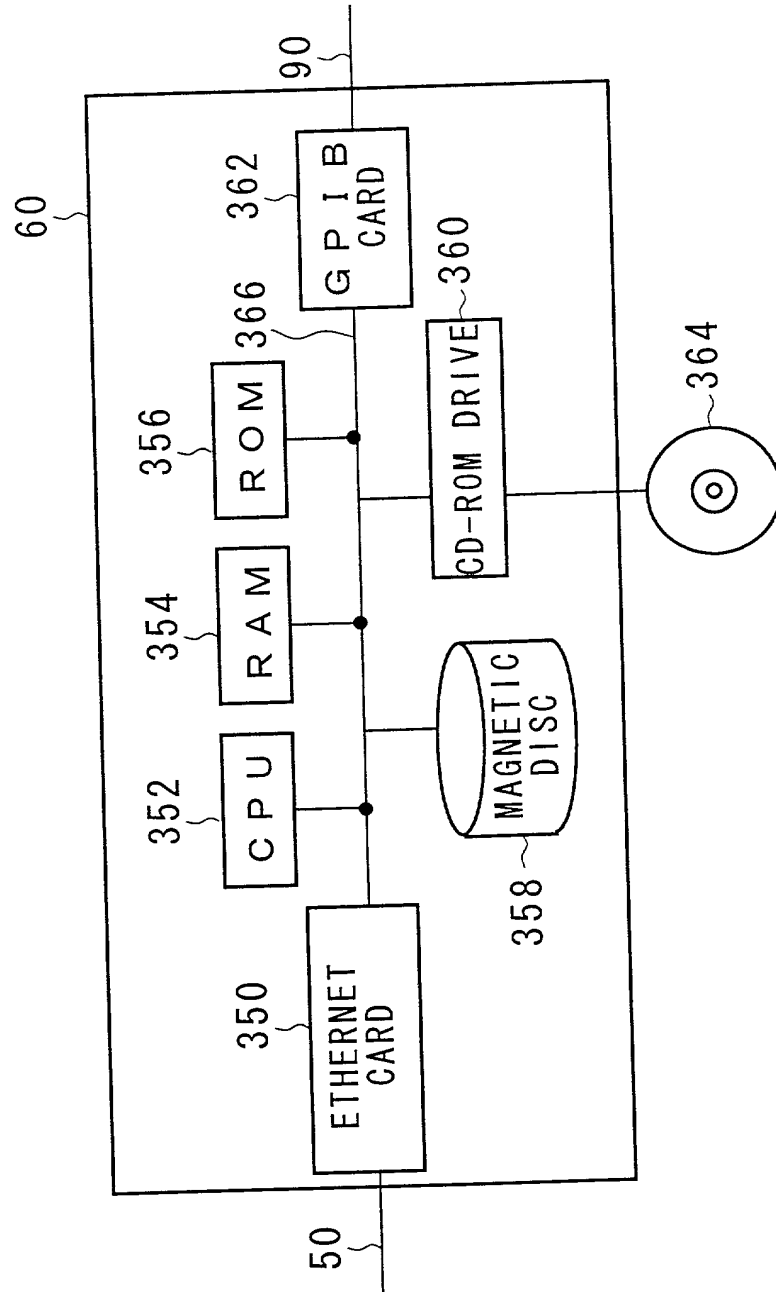


FIG . 14

```

public class program
{
    O BWmeasurement obw;
    A C Pmeasurement acp;
    dataTransportation dt;
    program()
    {
P1: GENERATE O BW MEASUREMENT OBJECT → obw = new O BWmeasurement();
P2: GENERATE A C P MEASUREMENT OBJECT → acp = new A C Pmeasurement();
P3: GENERATE MEASUREMENT DATA TRANSFER OBJECT → dt = new dataTransportation();
P4: PARAMETER SETUP → { obw.setCenter(918573500);
                        obw.setSpan(4000000);
                        obw.setBandRatio(99);
                        acp.setChannelBandWidth(1250000);
                        acp.setChannelSpacing(1250000);
                        }
P5: PROCESS TO BE PERFORMED → public boolean execute()
                                {
                                    obw.startMeasurement();
                                    acp.startMeasuremnet();
                                    dt.transport(obw.getData());
                                    dt.transport(acp.getData());
                                    return True;
                                }
P6: TRANSFER MEASUREMENT DATA OBJECT → {
                                public static void main(strinf args[])
                                {
                                    program meas = new program();
                                    meas.execute();
                                }
                                }
}

```

FIG . 15

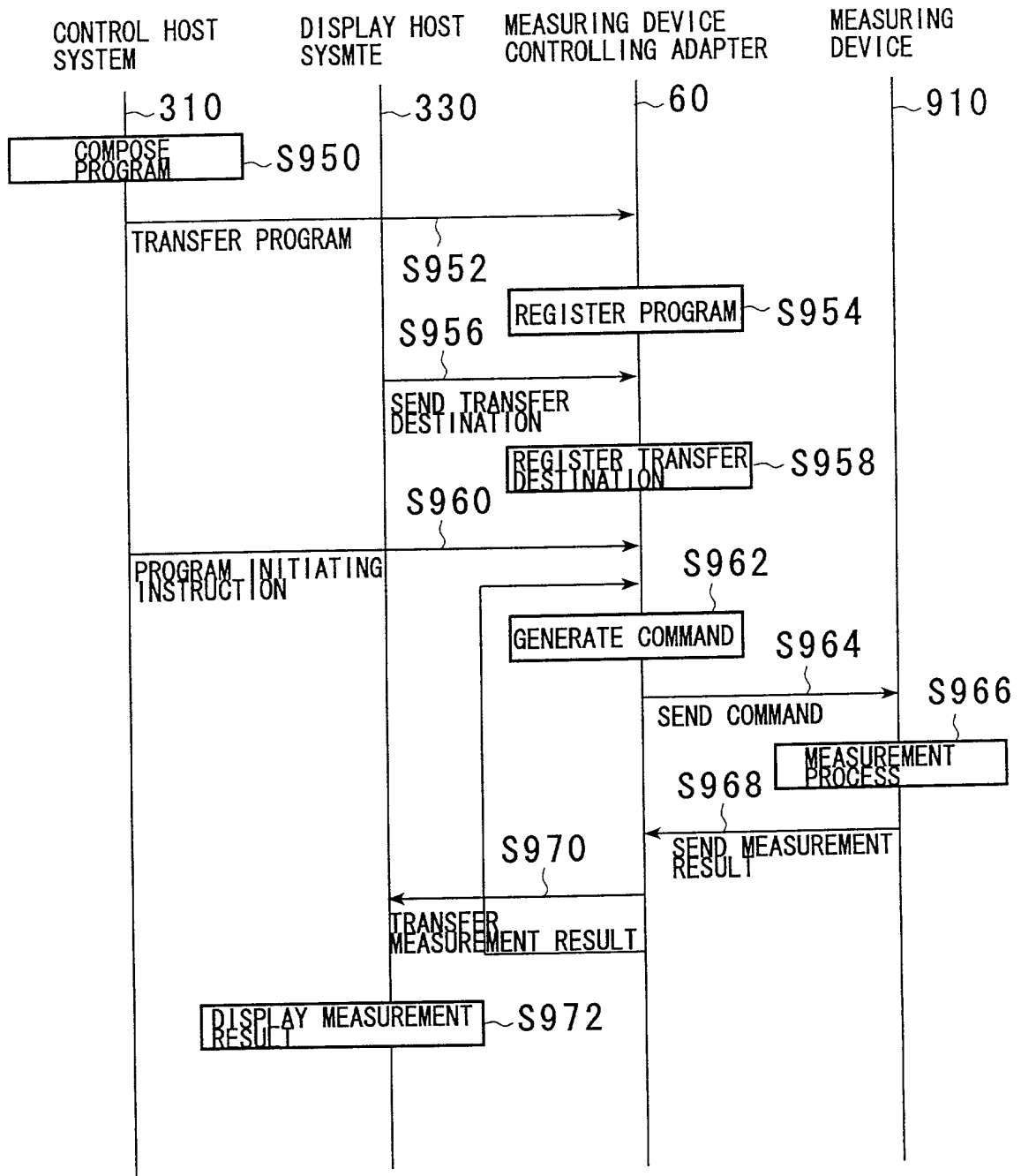


FIG. 16

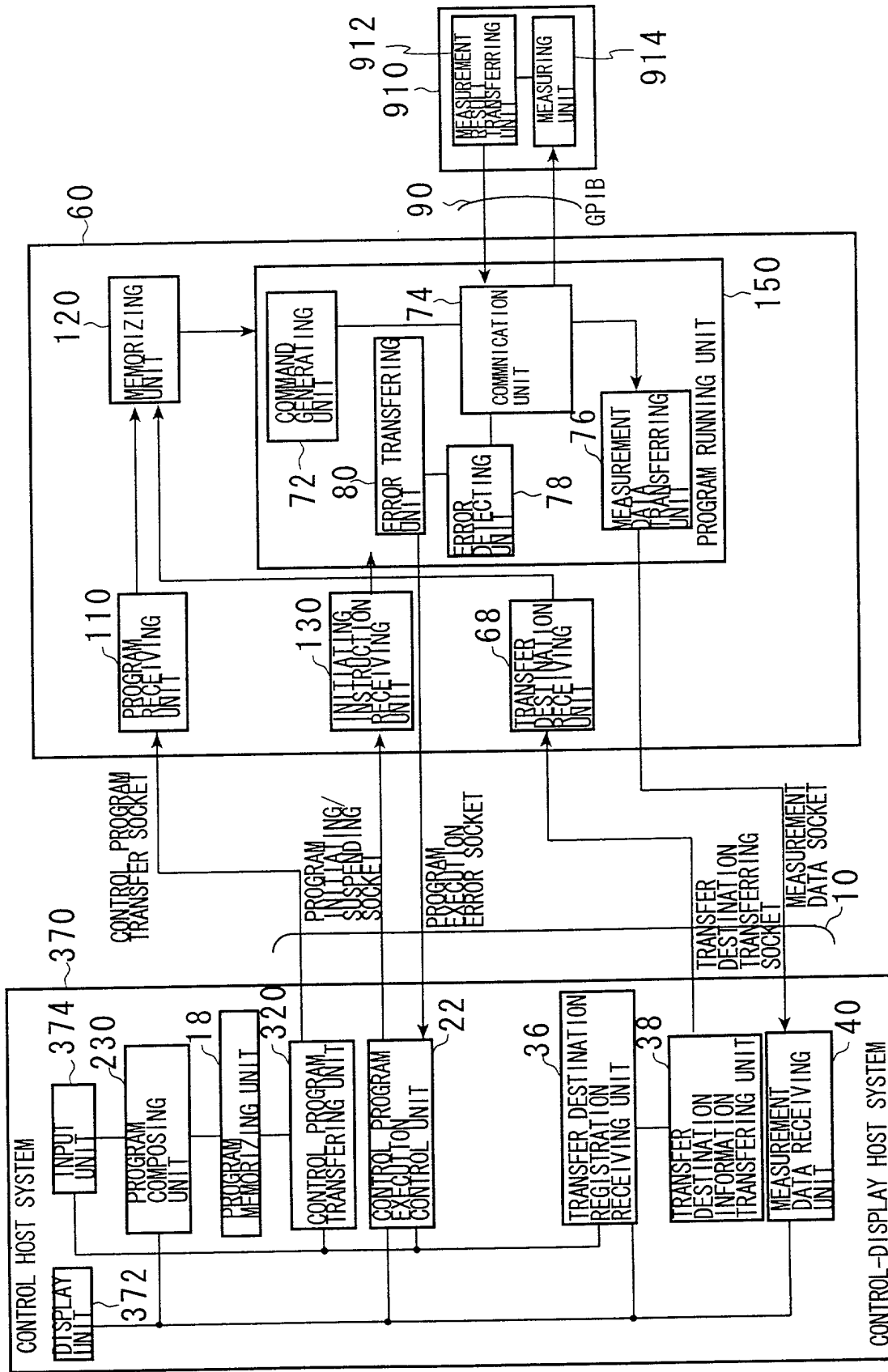


FIG. 17

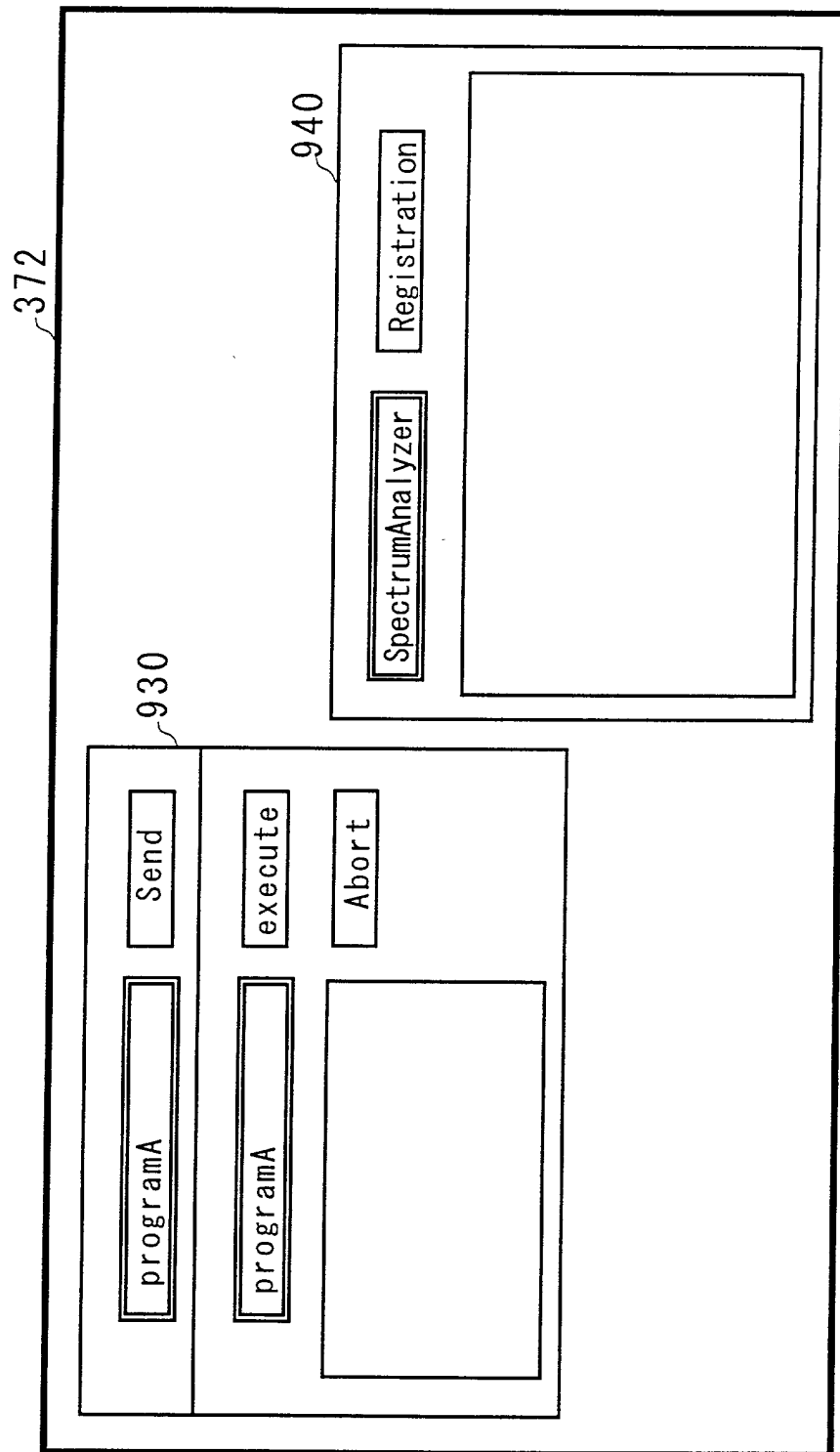


FIG . 18